REMARKS

With this response, claims 1, 3-16, 18-25, 27-33, and 37-41 are pending in the present application. With this response, claim 39 has been amended. For the reasons given below, it is submitted that this application is in condition for allowance. Thus, continued examination of the present application as amended is hereby requested.

Double Patenting

On page 2, the Office Action rejects claims 38 and 39 as being substantial duplicates of one another. While Applicants do not agree, claim 39 has been amended in order to expedite prosecution. Applicants therefore request that this rejection be withdrawn.

Omissions, Errors, Inconsistencies, and/or Ambiguities in Office Action

As discussed below, the Office Action contains certain omissions, errors, inconsistencies, and/or ambiguities which make it difficult for Applicants to respond effectively and precisely. Applicants respectfully requests that these be addressed if another Office Action is issued.

Rejection of Claims 1, 3-5, 7-9, 11-16, 18-20, 22-24, 27-33, and 37-41 Under 35 USC § 103(a)

On pages 2-9, the Office Action rejects claims 1, 3-5, 7-9, 11-16, 18-20, 22-24, 27-33, and 37-41 as being unpatentable over U.S. Patent No. 7,013,297 to Miksovsky (hereinafter 'Miksovsky') in view of U.S. Patent No. 7,296,256 to Liu et al. (Liu). Applicants respectfully traverse the rejection. Miksovsky and Liu are discussed first, followed by the Office Action's rejections of claims 1, 3-5, 7-9, 11-16, 18-20, 22-24, 27-33, and 37-41.

A. Miksovsky

Miksovsky discloses an expert system 308 (e.g., software) that receives intentions 324, or goals, of a user interface designer from an application written by a programmer and generates user interfaces in accordance with those intentions 324. Miksovsky, col. 1, l. 12-20. The system

generates the user interfaces from a set of programmatic rules, which are based on knowledge of experts in the field of user interfaces as embodied in guidelines, conventions, and principles of user interface design. *Id.*

According to Mikovsky, the expert system 308 begins by receiving a user interface goal or intention 324 (hereinafter 'interface intention') from an application 306 written by a programmer. Mikovsky, col. 8, l. 54-57, col. 9, l. 5-7, and Fig. 3A. Examples of interface intentions 324 are, for example, having the user supply a single string of text and having the user supply a single number. Mikovsky, col. 9, l. 23-39. The expert system 308 then receives parameters 326 from the application 306 or programmer. Mikovsky, col. 8, l. 57-61, col. 9, l. 40-48, col. 12, l. 39-45 and Fig. 4. Example parameters 326include the text of the question or instructions the programmer would like to offer the user and the choices from which the user is expected to make a selection. Mikovsky, col. 9, l. 49 through col. 10, l. 4. The expert system 308 may also consider other, external factors. Mikovsky, col. 8, l. 59-61 and col. 10, l. 5-8. Once the user interface intention(s) 324 and parameter(s) 326 are received, the expert system 308 may generate an appropriate user interface 310. Mikovsky, col. 8, l. 61-67. The user interface may then be invoked by the application 306 to interact with the user via an input device 314 and an output device 316. Mikovsky, col. 8, l. 63 through col. 9, l. 4, col. 12, l. 50-57, and FIG. 3A.

The expert system 308 is implemented as a collection of code modules or components 334-338. Mikovsky, col. 11, l. 21-23. Each code module is designed to generate an appropriate user interface for a single user interface goal from a group of available user interface templates. Mikovsky, col. 11, l. 23-27, and FIG. 3C. For example, choose component 334 in FIG. 3C corresponds to the intention 324 for helping a user to choose an item from a list 330. Mikovsky, col. 11, l. 47-48. The choose component 334 is invoked by the application 306 using a particular programmatic form, such as a function call and the appropriate parameters, such as a list of items 326A from which a user is to choose. Mikovsky, col. 11, l. 48-54.

B. Liu

Liu discloses combining "inference techniques with queuing models to automate the process of performance modeling and optimization of IT systems[, such as the e-business service structure

depicted in FIG. 4,] and applications." Liu, col. 2, l. 36-39. Liu "provides an end-to-end self-tuning, and flexible method that allows the building of performance models [] based on the system monitoring information." Liu, col. 4, l. 14-17. The model may then "provide insights of the capabilities of a Website, as well as a better understanding of the trade-offs between scalability, [Quality of Service (QoS)], capacity cost, operations risk, etc." and, therefore, plan and develop better IT solutions. Liu, col. 4, l. 20-29.

Liu builds a network model, such as the one depicted in FIG. 5, by obtaining the parameters of the service demands for each class at each generic server in the network model. Liu, col. 64-67. These server time parameters may be inferred from the most common measurements, such as the system throughput, utilization of the servers, and end-to-end response times. Liu, col. 8, l. 14-17. The methodology to infer these service parameters includes two aspects. Liu, col. 8, l. 25-26. First, the parameters can be inferred "if the underlying network has closed-form performance expressions (or good approximation[])." Liu, col. 8, l. 26-32. The use of close-form equations is depicted in col. 8, l. 34-57 and col. 9, l. 5-27 of Liu. Second, "if closed-form expressions or analytic approximations are not available, the present invention may rely on discrete-event simulator 611 together with a set of meta-heuristic search methods 613 including various tabu search and/or simulated annealing algorithms to search for the optimal set of parameters." Liu, col. 9, l. 37-42. The model may be build by, for example, module 603 in FIG. 6.

"Once a valid performance model has been established, the above closed-form expressions or the recursive mean-value analysis algorithm can be used to predict performance, optimize existing IT infrastructure, and to suggest cost-effective architecture design through deployment and operations. Those functions may be implemented in module 605 in FIG. 6." Liu, col. 10, l. 30-36.

C. Claims 1, 3-5, 7-9, 11-13, 40, and 41.

Regarding claim 1, Mikovsky and Liu taken either singly or in any reasonable combination, do not disclose or render obvious the claimed invention for at least the following four reasons.

1. The References do not disclose or suggest "changing parameters of the graphical model that are inconsistent with the at least one code generation goal," as recited in claim 1.

First, Mikovsky fails to disclose "changing parameters of the graphical model that are inconsistent with the at least one code generation goal," as recited in claim 1. In rejecting claim 1, the Office Action aligns almost two columns of Mikovsky, namely col. 9, l. 5 to col. 10, l. 70, with "changing parameters of the graphical model that are inconsistent with the at least one code generation goal." Office Action, pg. 3. Applicants disagree.

In contrast to claim 1, as Applicants discussed above, Mikovsky discloses an expert system 308 which receives a user interface intention 324 from an application 306 written by a programmer (Mikovsky, col. 9, l. 5-39), receives parameters 326 from the application 306 (Mikovsky, col. 9, l. 39 through col. 10, l. 4), and, optionally, considers external factors (Mikovsky, col. 10, l. 5-62). See also, Mikovsky, col. 8, l. 54-61, FIG. 3A, and FIG. 4. With respect to the received parameters, in order to be invoked properly, the expert system 308 may require the application 306 to provide certain parameters, depending on the user interface intention 324, and give the application 306 the option to specify additional parameters. Mikovsky, col. 11, l. 29-35. Once these **parameters 326 are received** and external factors considered, **an appropriate user interface is generated**, based on the received parameters 326 and external factors. Mikovsky, col. 8, l. 61-67 and col. 11, l. 29-35. **However, generating a user interface based on received parameters 326 is not equivalent to "changing parameters of the graphical model that are inconsistent with the at least one code generation goal," as recited in claim 1**. Therefore, Mikovsky does not disclose "changing parameters of the graphical model that are inconsistent with the at least one code generation goal," as recited in claim 1.

Furthermore, Liu fails to overcome the deficiencies of Mikovsky. As Applicants discussed above, Liu discloses the creation of performance models of IT systems so that those systems can be analyzed and refined. Liu, col. 4, l. 20-29. Liu builds a network model, such as the one depicted in FIG. 5, by obtaining the parameters of the service demands for each class at each generic server in the network model. Liu, col. 64-67. The creation of the performance models in Liu is not equivalent to "changing parameters of the graphical model that are inconsistent with the at least one

code generation goal," as recited in claim 1. The combination of Mikovsky and Liu, therefore, fails to teach or render obvious "changing parameters of the graphical model that are inconsistent with the at least one code generation goal," as recited in claim 1.

2. The Office Action does not address "at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in claim 1.

Second, the Office Action does not address "at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in claim 1. In particular, the Office Action relies on Mikovsky to teach every feature of claim 1 except for "the graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations, the graphical model being capable of simulation based on the equations." Applicants agree that Mikovsky does not teach these features. However, in addition to not teaching these features of claim 1, Mikovsky fails to disclose "at least one code generation goal relating to a characteristic of the code to be generated from the graphical model."

In rejecting claim 1, the Office Action discusses a limitation of "each embedded code relating to a characteristic of code to be generated from the graphical interface" and states that it corresponds to col. 11, l. 65 through col. 12, l. 12 of Mikovsky. Office Action, pg. 3. However, claim 1 does not recite "each embedded code relating to a characteristic of code to be generated from the graphical interface." Thus, for the purposes of this response, the Office Action is assumed to have aligned the recited "at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" of claim 1 with col. 11, l. 65 through col. 12, l. 12 of Mikovsky.

In contrast to claim 1, Mikovsky does not disclose "at least one code generation goal relating to a characteristic of the code to be generated from the graphical mode," as recited in claim 1. Instead, as Applicants discussed above, Mikovsky discloses an expert system 308 which is implemented as a collection of code modules or components 334-338, where each code module is designed to generate an appropriate user interface. Mikovsky, col. 11, l. 21-27. Individual code modules are invoked by an application 306 using a particular programmatic form, such as a function

call, and the appropriate parameters. Mikovsky, col. 11, l. 48-54. Once the expert system 308 has determined what sort of user interface is appropriate, a user interface is generated and returned to the application 306. Mikovsky, col. 11, l. 65 through col. 12, l. 1. Where, for example, the expert system 308 is operated in a object-oriented environment, the user interface may be dynamically created during program execution, passed between program functions, have its member functions invoked, and have these member functions return results to the invoking program. Mikovsky, col. 12, l. 1-10.

However, displaying a dynamically created user interface in an object-oriented environment, where the user interface is invoked by an application 306 is not equivalent to "at least one code generation goal relating to a characteristic of the code to be generated from the graphical mode," as recited in claim 1. Therefore, Mikovsky does not disclose "at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1.

Liu does not overcome the failings of Mikovsky. Instead, Liu discloses the creation of performance models of IT systems so that those systems can be analyzed and refined. Liu, col. 4, l. 20-29. Liu builds a network model, such as the one depicted in FIG. 5, by obtaining the parameters of the service demands for each class at each generic server in the network model. Liu, col. 64-67. Thus, the creation of the performance models in Liu is not equivalent to "at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in claim 1. Hence, the combination of Mikovsky and Liu fail to teach or render obvious "at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in claim 1.

Furthermore, Applicants request that the recited "at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" feature of claim 1 be properly and precisely addressed in the next Office Action, if one is issued.

3. The Office Action does not address "prompting a user to specify at least one code generation goal from a plurality of code generation goals," as recited in claim 1.

Third, the Office Action does not address "prompting a user to specify at least one code generation goal from a plurality of code generation goals [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model," as recited in claim 1. In particular, the Office Action relies on Mikovsky to teach every feature of claim 1 except for "the graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations, the graphical model being capable of simulation based on the equations." Applicants agree that Mikovsky does not teach these features. However, in addition to not teaching these features of claim 1, Mikovsky fails to disclose "prompting a user to specify at least one code generation goal from a plurality of code generation goals [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model."

In rejecting claim 1, the Office Action discusses a limitation of "prompting a user to specify the embedded code from a plurality of embedded code" and states that it corresponds to Col. 11, l. 37-65 of Mikovsky. Office Action, pg. 3. However, claim 1 does not recite "prompting a user to specify the embedded code from a plurality of embedded code." Thus, for the purposes of this response, the Office Action is assumed to have aligned the recited "prompting a user to specify at least one code generation goal from a plurality of code generation goals [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model," of claim 1 with the discussion of FIG. 3C in Col. 11, l. 37-65 of Mikovsky.

In contrast to claim 1, Mikovsky does not disclose "prompting a user to specify at least one code generation goal from a plurality of code generation goals [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model," as recited in claim 1. Instead, as Applicants discussed above, Mikovsky discloses an expert system 308 which is implemented as a collection of code modules or components 334-338, where each code module is designed to generate an appropriate user interface. Mikovsky, col. 11, l. 21-27. Individual code modules are invoked by an application 306 using a particular programmatic form, such as a function call, and the appropriate parameters. Mikovsky, col. 11, l. 48-54.

However, an application 306 invoking an expert system 308, via particular programmatic forms, which generates an appropriate user interface, is not equivalent to

"prompting a user to specify at least one code generation goal from a plurality of code generation goals .... [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model," as recited in claim 1. Therefore, Mikovsky does not disclose "prompting a user to specify at least one code generation goal from a plurality of code generation goals [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model," as recited in claim 1.

Liu does not overcome the failings of Mikovsky. Instead, Liu discloses the creation of performance models of IT systems so that those systems can be analyzed and refined. Liu, col. 4, l. 20-29. Liu builds a network model, such as the one depicted in FIG. 5, by obtaining the parameters of the service demands for each class at each generic server in the network model. Liu, col. 64-67. The creation of the performance models in Liu is not equivalent to the "code generation goal" of claim 1 which "relat[es] to a characteristic of the code to be generated from the graphical model." **Therefore, creating a performance model, as Liu discloses, is not equivalent to "prompting a user to specify at least one code generation goal from a plurality of code generation goals [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model," as recited in claim 1**. Hence, the combination of Mikovsky and Liu fail to teach or render obvious "prompting a user to specify at least one code generation goal from a plurality of code generation goals [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model," as recited in claim 1.

Furthermore, Applicants request that the recited "prompting a user to specify at least one code generation goal from a plurality of code generation goals [where] the at least one code generation goal relat[es] to a characteristic of the code to be generated from the graphical model" feature of claim 1 be properly and precisely addressed in the next Office Action, if one is issued.

4. The References do not disclose or suggest "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment," as recited in claim 1.

Fourth, Mikovsky fails to disclose "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment," as recited in claim 1. In rejecting claim 1, the Office Action aligns almost all of col. 9 and 10 of Mikovsky, namely col. 9, l. 5 to col. 10, l. 70, with "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment." Office Action, pg. 3. Applicants disagree.

In contrast to claim 1, as Applicants discussed above, Mikovsky discloses an expert system 308 which receives a user interface goal or intention 324 from an application 306 written by a programmer (Mikovsky, col. 9, l. 5-39), receives parameters 326 from the application 306 (Mikovsky, col. 9, l. 39 through col. 10, l. 4), and, optionally, considers external factors (Mikovsky, col. 10, l. 5-62). See also, Mikovsky, col. 8, l. 54-61, FIG. 3A, and FIG. 4. Once these **parameters 326 are received** and external factors considered, **an appropriate user interface is generated**, based on the received parameters 326 and external factors,. Mikovsky, col. 8, l. 61-67 and col. 11, l. 29-35. Where, for example, the expert system 308 is operated in a object-oriented environment, the user interface may be dynamically created during program execution and displayed as an object within the environment. Mikovsky, col. 12, l. 1-10.

However, generating a user interface based on received parameters 326 and displaying the user interface is an object-oriented environment, is not equivalent to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment," as recited in claim 1. Therefore, Mikovsky does not disclose "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment," as recited in claim 1.

Furthermore, Liu fails to overcome the deficiencies of Mikovsky. Liu discloses the creation of performance models of IT systems so that those systems can be analyzed and refined. Liu, col. 4, l. 20-29. Liu builds a network model, such as the one depicted in FIG. 5, by obtaining the parameters of the service demands for each class at each generic server in the network model. Liu, col. 64-67. The creation of the performance models in Liu is not equivalent to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment," as recited in claim 1. The combination of Mikovsky and Liu, therefore,

fails to teach or render obvious "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment," as recited in claim 1.

Therefore, for the above four reasons, Mikovsky and Liu, taken either singly or in any reasonable combination, fail to disclose or render obvious claim 1. Applicants believe that claim 1 is allowable and respectfully request that the above rejection of claim 1 be withdrawn and that claim 1 be allowed.

Dependent claims 3-5, 7-9, 11-13, 40, and 41 depend on claim 1 and are believed to be allowable for at least the same reasons as above. Therefore, Applicants respectfully request that the above rejection of claims 3-5, 7-9, 11-13, 40, and 41 be withdrawn and that claims 3-5, 7-9, 11-13, 40, and 41 be allowed.

D. Claims 14, 16, and 30-33

Independent claims 14, 16, and 30-33 recite subject matter similar to that recited in claim 1, which Applicants believe is allowable over Mikovsky in view of Liu. Therefore, Applicants believe claims 14, 16, and 30-33 are allowable for at least the reasons set forth above. Applicants respectfully request that the above rejection of claims 14, 16, and 30-33 be withdrawn and that claims 14, 16, and 30-33 be allowed.

Claim 14 recites "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations," which is similar to "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations" as recited in claim 1. Claim 14 recites "the graphical model being capable of simulation based on the equations," which is similar to "the graphical model being capable of simulation based on the equations" as recited in claim 1. Claim 14 recites "the code generation goal being used to generate embedded code from the graphical model, the code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code generation goal being used to generate

embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 14 recites "generating embedded code in accordance with the code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 16 recites "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations," which is similar to "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations" as recited in claim 1. Claim 16 recites "the graphical model being capable of simulation based on the equations," which is similar to "the graphical model being capable of simulation based on the equations" as recited in claim 1. Claim 16 recites "the acquired at least one code generation goal being used to generate embedded code from the graphical model, the acquired at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 16 recites "generating embedded code in accordance with the acquired at least one code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 30 recites "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations," which is similar to "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations" as recited in claim 1. Claim 30 recites "the graphical model being capable of simulation based on the equations," which is similar to "the graphical model being capable of simulation based on the equations" as recited in claim 1. Claim 30 recites "the acquired at least one code generation goal being used to generate embedded code from the graphical model, the acquired at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code

generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 30 recites "one or more instructions for generating embedded code in accordance with the acquired at least one code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 31 recites "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations," which is similar to "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations" as recited in claim 1. Claim 31 recites "the graphical model being capable of simulation based on the equations," which is similar to "the graphical model being capable of simulation based on the equations" as recited in claim 1. Claim 31 recites "the code generation goal being used to generate embedded code from the graphical model, the code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 31 recites "one or more instructions for generating embedded code in accordance with the code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 32 recites "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations," which is similar to "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations" as recited in claim 1. Claim 32 recites "the graphical model being capable of simulation based on the equations," which is similar to "the graphical model being capable of simulation based on the equations" as recited in claim 1. Claim 32 recites "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at

least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 30 recites "one or more instructions for generating embedded code in accordance with the at least one code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 33 recites "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations," which is similar to "graphical model representing a dynamic system having time-changing behavior modeled with differential, difference, and/or algebraic equations" as recited in claim 1. Claim 33 recites "the graphical model being capable of simulation based on the equations," which is similar to "the graphical model being capable of simulation based on the equations" as recited in claim 1. Claim 33 recites "a process for preparing a graphical model for a code generation process for creating code based on the graphical model and at least one code generation goal, wherein the at least one code generation goal relates to a characteristic of the code," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 33 recites "wherein the computer program generates code in compliance with the at least one code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

E. Claims 15, 18-20, 22-24, 27-29 and 37-41.

Dependent claims 15, 18-20, 22-24, 27-29 and 37-41 depend from claims 14, 16, or 33, and therefore are believed to be allowable for at least the same reasons as above. Therefore, Applicants respectfully request that the above rejection of claims 15, 18-20, 22-24, 27-29 and 37-41 be withdrawn and that claims 15, 18-20, 22-24, 27-29 and 37-41 be allowed.

Rejection of Claims 10 and 25 Under 35 USC § 103(a)

On page 9, the Office Action rejects claims 10 and 25 as being unpatentable over Miksovsky in view of Liu and in further view of U.S. Patent No. 6,851,105 to Coad et al. (Coad). Applicants respectfully traverse the rejection.

Dependent claims 10 and 25 depend on claims 1 and 16, respectively, and therefore are believed to be allowable for at least the same reasons as above. Therefore, Applicants respectfully request that the above rejection of claims 10 and 25 be withdrawn and that claims 10 and 25 be allowed.

Rejection of Claims 6 and 21 Under 35 USC § 103(a)

On pages 9-11, the Office Action rejects claims 6 and 21 as being unpatentable over Miksovsky in view of Liu and in further view of U.S. Patent No. 5,339,433 to Frid-Nielsen et al. (Frid-Nielsen). Applicants respectfully traverse the rejection.

Dependent claims 6 and 21 variously depend on claims a and 16, respectively, and therefore are believed to be allowable for at least the same reasons as above. Therefore, Applicants respectfully request that the above rejection of claims 6 and 21 be withdrawn and that claims 6 and 21 be allowed.

Conclusion

All of the stated grounds of rejections and objections have been properly traversed, accommodated, or rendered moot. Applicants therefore respectfully request that the Examiner reconsider all presently outstanding rejections and objections and that they be withdrawn. Applicants believe that a full and complete reply has been made to the outstanding Office Action and, as such, the present application is in condition for allowance. If the Examiner believes, for any reason, that personal communication will expedite prosecution of this application, the Examiner is hereby invited to telephone the undersigned at the number provided.

Prompt and favorable consideration of this Amendment is respectfully requested.

Dated: October 22, 2009

Respectfully submitted,

By _____

Michael A. Sartori, Ph.D.
Registration No.: 41,289
Kyle D. Petaja
Registration No.: 60,309
VENABLE LLP
P.O. Box 34385
Washington, DC 20043-9998
(202) 344-4000
(202) 344-8300 (Fax)
Attorney/Agent For Applicant

#1060886v1